Santander Consumer Bank
ul. Legnicka 48 B
54-202 Wrocław
phone: 71 387 74 04
wsparcie@santanderconsumer.pl

Instructions for Integrating E-commerce Systems with Santander Consumer Bank’s **eRaty** Online Installment Sales System

2024-04-15 (Version 5.3)

Contents

1. Introduction 3

2. Glossary 4

3. User scenarios 4

3.1 Submission of a loan application for the purchase of goods and services and its verification 4

3.1.1 Preliminary assumptions 4

3.1.2 Process flow – basic scenario 4

3.1.3 End result 5

3.1.4 Alternative scenarios 5

4. POST message 6

5. WebService 9

5.1 Installation and use of the certificate 10

5.2 Description of service methods 12

5.2.1 IsActive 12

5.2.2 GetApplicationState 13

5.2.3 GetFinancialData 16

5.6 Operation errors 19

6. Additional options 20

6.1 Information on the Store’s website 20

6.2 Installment simulator 21

6.3 Link generator 23

7. How to carry out the process of integration with the Bank 24

8. Appendices 26

1. Introduction

This document is a manual for connecting an e-commerce system with Santander Consumer Bank’s eRaty system, which allows Customers to apply for and conclude loan agreements for the purchase of goods and services.

ABOUT THIS DOCUMENT

The document describes:

- the capabilities provided by the integration with the eRaty system,
- how to choose the right integration method,
- how to enable Customers to pay for goods in installments, and in particular how to integrate an online store with the eRaty system,
- how to test the functionality of an online store after integration with the eRaty system, and
- how to receive status messages for loan applications for the purchase of goods and services, and how to translate the received status messages into specific actions on the Store’s part (optional functionality).

The eRaty system is designed both for Stores that want to provide their Customers with permanent access to installment payments, and those in which installment payments will be used occasionally, at the Customer’s request, through the so-called Link Generator.

In these Instructions, we have described the integration, which is characterized by the fact that the entire process of concluding an agreement is done on the Bank’s part. The Store redirects the Customer to the Bank’s request via a POST message. Throughout the process, the Store is informed about the application status via e-mail communication and the provided banking application (Panel Sklep), and additionally it is possible to communicate via WebService (which the Store implements integration with on its part).

For Stores, we also provide a Loan Simulator, which allows you to simulate calculation of the amount of installment and costs for a loan for the purchase of goods and services, before the Customer even starts filling in the application.

INTEGRATION WITH POPULAR STORE SYSTEMS

The eRaty system is already integrated with many store systems available on the market.
Their list is available at: https://www.eraty.pl/systemy-sklepowe/

ABOUT THE LOAN APPLICATION PROCESS

A simplified description of the basic process is as follows:

1. The Store sends a POST message and redirects the Customer to the Bank’s loan application form.
2. The Customer completes the data in the application.
3. The Customer sends the application for verification.
4. The Bank verifies the Customer’s application and issues a positive loan decision.
5. The Customer confirms their identity (by a wire transfer for PLN 1 or with an SMS code – if such option is available to the Customer).
6. The Customer attaches document scans – if required.

7. The Bank verifies the Customer’s transfer/SMS code and the attached document scans (if required).
8. The Bank grants the loan.
9. The Bank activates the Customer’s agreement and transfers the funds to the Store.

Throughout the process, the Customer is kept informed of the status of their application through messages sent by the Bank to the Customer’s e-mail address provided in the application. The e-mails also include a link that allows the Customer to return to the application at any time.

2. Glossary

The terms used in this document have the following meanings:

Bank – Santander Consumer Bank S.A., a credit service provider that provides and processes the installment application and verifies the Customer’s creditworthiness.
eRaty – the Bank’s installment sales system, through which the Customer can submit an application and conclude a loan agreement.
Customer – a person who intends to finance their purchase in the Store by means of a loan for the purchase of goods and services.
Account Manager – a representative of the Bank, dedicated to supporting the cooperation with the Partner.
Panel Sklep – the Bank’s application that enables the exchange of information between the Bank and the Store about the status of loan applications submitted by Customers and is used to confirm the availability of goods.
Partner – an entrepreneur who has entered into a cooperation agreement with the Bank on installment sales of goods and/or services.
Point of Sale (PoS) – a Store parameterized in the Bank’s records based on the registration and address data provided in the cooperation agreement.
Store – Partner’s online store that integrates with the eRaty system and is the direct supplier of the goods and/or services ordered by the Customer.

3. User scenarios

3.1 Submission of a loan application for the purchase of goods and services and its verification

3.1.1 Preliminary assumptions

- The Customer has placed an order in the Store.
- The Customer has chosen a loan for the purchase of goods and services as a payment method.

3.1.2 Process flow – basic scenario

1. The Store initiates the loan application process:
    a. The Store sends a POST message/loan application form to the Bank, with all mandatory parameters.

   b. The Bank receives the message.
   c. The Customer is redirected to the loan application form.

2. The Bank calculates the amount of the loan installment based on the received parameters, and presents the result to the Customer on the first screen of the application, i.e. the Calculator screen. On the Calculator screen, the Customer can select an installment product (1 to 3 installment products can be presented in the Calculator) and modify its parameters (number of installments, date of first repayment, down payment, insurance).
3. In further steps, the Customer fills in all the data required in the loan application. The application will be populated with the data entered by the Customer at the Store level, which was transferred to the eRaty system as POST parameters.
4. After completing all the data, the Customer sends the application for verification by the Bank.
5. The Customer is verified positively.
6. The Bank presents an agreement to the Customer.
7. The Customer accepts the agreement.
8. The Customer confirms their identity by making a verification transfer for PLN 1 or providing an SMS code (if such method of confirming identity is available to the Customer), and attaches document scans (if required).
9. The Bank verifies the transfer or SMS code (and document scans, if required).
10. The Bank has positively verified the Customer’s identity (and document scans, if required).
11. The Customer is redirected from the loan application to the Store to the link provided in the wniosekZapisany parameter (from the POST message).
12. The scenario ends.

3.1.3 End result

- The Bank has granted the Customer a loan for the purchase of goods and services.
- The Store has issued the goods to the Customer.
- The agreement has been activated.
- The funds for the goods have been transferred to the Store.

3.1.4 Alternative scenarios

5.a. The Customer is verified negatively.

6. The Customer is redirected from the loan application to the Store to the link provided in the wniosekAnulowany parameter (from the POST message).
7. The scenario ends.

10.a. The Bank has negatively verified the Customer’s identity.

11. The Customer is redirected from the loan application to the Store to the link provided in the wniosekAnulowany parameter (from the POST message).
12. The scenario ends.

10.b. The Customer has failed to meet the formal requirements or abandoned the loan application.

11. The Customer is redirected from the loan application to the Store to the link provided in the wniosekAnulowany parameter (from the POST message).
12. The scenario ends.

4. POST message

With the POST message, the Store forwards to the Bank the details of the Point of Sale and the Customer's purchase and thus initiates the loan application process. In the table below, we have described the parameters included in the message. At the integration stage, the Account Manager will forward to the Store the URL to which the Customer should be redirected. A POST message to this address should be sent from the Customer's browser.

Parameter name	**Required (YES/NO)**	**Parameter type and length**	**Parameter description**
numerSklepu	YES	NUMBER(7)	Point of Sale identification number. The number can be found in the attachment to the cooperation agreement on installment sales of goods and/or services concluded between the Partner and the Bank.
typProduktu	YES	NUMBER(2)	The variable used within the eRaty system, always takes the value "0" (zero).
nrZamowieniaSklep	YES	VARCHAR2(30)	Order number in the Store's system. Each order should have a unique number in the Store. Re-using an order number for another purchase will result in an error when redirecting to the loan application.
idTowaru(n)	YES	VARCHAR2(20)	ID of the nth product in the Store / ID of the product in the Store's database.
nazwaTowaru(n)	YES	VARCHAR2(50)	Name of the nth product in the Store / name of the product in the Store's database. Product names must not contain apostrophes, e.g. 32" TV set, because this will result in incorrect closing of the <input> tag.
wartoscTowaru(n)	YES	DOUBLE(11,2)	Gross price of the nth product in the Store / price of the product in the Store's database. The final price, that is, including VAT and any discounts applied by the Partner. The price format must not include spaces or letters. The price format can include digits and a comma or period.
liczbaSztukTowaru(n)	YES	DOUBLE(7,3)	Number of pieces of the nth product in the Store that the Customer has ordered.
jednostkaTowaru(n)	YES	VARCHAR2(10)	Unit of measure of the quantity of the nth product in the Store, such as pieces, kg, cm, liters, etc.
wartoscTowarow	YES	any	Total value of the goods in the Customer's order, i.e.: (pieces of the nth prod.) * (price of the nth prod.) + (pieces of n+1 prod.) * (price of n+1 prod.) + ... = total value of goods. If the total value of the goods is greater than the value produced by this variable, then the eRaty system reads this as a purchase with a discount on the total cart. The price format must not include spaces or letters. The price format can include digits and a comma or period.

Parameter name	**Required (YES/NO)**	**Parameter type and length**	**Parameter description**
liczbaSztukTowarow	YES	any	Number of items in the Customer’s order, i.e.: n * Product = total number of items
sposobDostarczeniaTowaru	YES	VARCHAR2(50)	Method (name) of delivery of goods (e.g. courier, mail, etc.).
char	YES	[ISO, UTF, WIN]	Set of character encoding sent by the Store. Only the following values are available: ISO, UTF, WIN (only these should be sent). They correspond to these values: ISO-8859-2, UTF-8, WINDOWS-1250.
wniosekZapisany	YES	any	Address of the website to which the Customer is to be redirected after concluding the loan agreement.
wniosekAnulowany	YES	any	Address of the website to which the Customer is to be redirected after abandoning the loan application for the purchase of goods and services or after the Bank refuses to grant the loan.
pesel	NO	VARCHAR2(11)	Customer’s PESEL No.
imie	NO	VARCHAR2(35)	Customer’s first name.
nazwisko	NO	VARCHAR2(35)	Customer’s last name.
email	NO	VARCHAR2(64)	Customer’s e-mail address.
telKontakt	NO	VARCHAR2(9)	Customer’s phone number.
ulica	NO	VARCHAR2(50)	Customer’s street address.
nrDomu	NO	VARCHAR2(7)	Customer’s house number.
nrMieszkania	NO	VARCHAR2(7)	Customer’s apartment number.
miasto	NO	VARCHAR2(50)	Customer’s city/town.
kodPocz	NO	NUMBER(6)	Customer’s zip code. Zip code format: xx-xxx, must include a dash after the first two digits, e.g. 22-222.
blokadaWplaty	NO	NUMBER(1)	Ability to block editing of down payment by the Customer in the application: 0 – the Customer can edit the down payment 1 – the Customer cannot edit the down payment If the value of the field is not submitted, the value set during the registration of the Store in the Bank’s system will be used.

Parameter name	Required (YES/NO)	Parameter type and length	Parameter description
liczbaRat	No	NUMBER(3)	Default number of installments, which should be marked for the product after the Customer enters the application. If the field value is not submitted, the maximum number of installments available for the loan product will be used.

In the POST message, the Customer's details are not required, as the Customer can provide them themselves in the application. However, it is worth noting that the Customer details received via POST will be populated in the application, which will significantly shorten the loan application process.
Importantly, the POST message may contain selected Customer details, those that the Customer provided on the Store's website, e.g. first name, last name and e-mail address.

Sample appearance of the submitted form

Customer purchase parameters:

Product name	Quantity	Price
Product A	1	12.00
Product B	2	134.87
Shipping cost	1	25.00

```
<form name="formularz_eRaty" action="https://wniosek.eraty.pl/formularz/" method="post">
<input name="idTowaru1" readonly="readonly" type="hidden" value="37">
<input name="nazwaTowaru1" readonly="readonly" type="hidden" value="Produkt A">
<input name="wartoscTowaru1" readonly="readonly" type="hidden" value="12.00">
<input name="liczbaSztukTowaru1" readonly="readonly" type="hidden" value="1">
<input name="jednostkaTowaru1" readonly="readonly" type="hidden" value="szt.">
<input name="idTowaru2" readonly="readonly" type="hidden" value="30">
<input name="nazwaTowaru2" readonly="readonly" type="hidden" value="Produkt B">
<input name="wartoscTowaru2" readonly="readonly" type="hidden" value="134.87">
<input name="liczbaSztukTowaru2" readonly="readonly" type="hidden" value="2">
<input name="jednostkaTowaru2" readonly="readonly" type="hidden" value="szt.">
<input name="idTowaru3" readonly="readonly" type="hidden" value="KosztPrzesylki">
<input name="nazwaTowaru3" readonly="readonly" type="hidden" value="Koszt Przesyłki">
<input name="wartoscTowaru3" readonly="readonly" type="hidden" value="25.00">
<input name="liczbaSztukTowaru3" readonly="readonly" type="hidden" value="1">
<input name="jednostkaTowaru3" readonly="readonly" type="hidden" value="szt.">
<input type="hidden" name="wartoscTowarow" value="306.74">
<input type="hidden" name="liczbaSztukTowarow" value="4">
<input type="hidden" name="numerSklepu" value="NUMER_SKLEPU_ERATY">
<input type="hidden" name="typProduktu" value="0">
<input type="hidden" name="sposobDostarczeniaTowaru" value="np. Przesyłka kurierska">
<input type="hidden" name="nrZamowieniaSklep" value="NR_ZAMOWIENIA_SKLEPU">
<input type="hidden" name="pesel" value="">
<input type="hidden" name="imie" value="test">
<input type="hidden" name="nazwisko" value="test">
<input type="hidden" name="email" value="email@klienta.pl">
<input type="hidden" name="telKontakt" value="2222222">
```

```
<input type="hidden" name="ulica" value="testowa">
<input type="hidden" name="nrDomu" value="22">
<input type="hidden" name="nrMieszkania" value="22">
<input type="hidden" name="miasto" value="test">
<input type="hidden" name="kodPocz" value="22-222">
<input type="hidden" name="blokadaWplaty" value="0">
<input type="hidden" name="liczbaRat" value="12">
<input type="hidden" name="char" value="ISO">
<input type="hidden" name="wniosekZapisany"
value="http://www.twoj-sklep.pl/strona_powrotu.php?wniosek=przyjety&id_zamowienie=">
<input type="hidden" name="wniosekAnulowany" value="http://www.twoj-
sklep.pl/strona_powrotu.php?wniosek=odrzucony&id_zamowienie=">
<input type="image" src="https://www.eraty.pl/images/bannery/zlozwniosek_white.png"
border="0" alt="Kontynuuj - Złóż wniosek o kredyt ratalny Santander Consumer Banku”
title="Złóż wniosek o kredyt ratalny Santander Consumer Banku">
</form>
```

Resubmitting the form with the same order number

After receiving a POST message from the Store with an order form, a check is performed to see if an application from the same Store and with the same order number as the one just sent already exists in the Bank’s databases:

- if such application DOES NOT exist, the Customer is redirected from the Store to a new application created for the order;
- if such application EXISTS, but it has a status that prevents its further processing, the Customer is redirected from the Store to a new application created for the renewed order;
- if such application EXISTS and has a status that enables its further processing, the content of the shopping cart sent in the form is additionally verified;
    - if the content of the shopping cart in the new order is THE SAME as in the order form based on which the existing application was created, the Customer is redirected from the Store to a link to continue the process for the active application;
    - if the content of the shopping cart in the new order is DIFFERENT than in the order form based on which the existing application was created, the Customer cannot continue processing the active application and the new application is not created, and the Bank returns an error message to the Store with the following content: “The application cannot be created. There is already an active application with order number <nr_zamowienia>”.

5. WebService

In addition to the simple integration based on the Store sending POST messages, the Bank also offers stores an extended integration model by providing the ProposalServiceHybrid WebService for the Store’s communication with the Bank. The active party here is the Store, which sends a request to the Bank and waits for a response.

The address for the service is:
https://api.santanderconsumer.pl/ProposalServiceHybrid
The WSDL is available at: https://api.santanderconsumer.pl/ProposalServiceHybrid?WSDL

The service is secured by:
– **login and password** – the access account is sensitive to incorrect login attempts; after 3 login attempts with the wrong password, the account is blocked; and
– **certificate** – the certificate has a limited validity; from time to time the Bank will provide a new certificate for installation.

5.1 Installation and use of the certificate

In order to establish communication with the WebService, it is required that the Store use a certificate. The certificate, the password to it and the authorities' "Finance For Home" certificates are provided to the Partner by the Account Manager.

Ways to implement connection with the certificate

Below are examples of how to implement a WebService call using a certificate provided by the Bank.
The first step is to install the certificate in .pfx format. After installing the certificate, in order to avoid problems with its use, the authority's certificate issued for the Bank should be whitelisted.
Once the certificate is installed correctly and the password is entered, it will be possible to open the WebService's WSDL page in the browser:
https://api.santanderconsumer.pl/ProposalServiceHybrid?WSDL

The next steps depend on the development environment used.

PHP

1. Use openssl to convert the .pfx certificate into a .pem certificate. Here, too, it will be necessary to use the certificate password.
   The command that turns certyfikat.pfx into certyfikat.pem is as follows:

```
openssl pkcs12 –in ścieżka_do_certyfikatu/certyfikat.pfx –out certyfikat.pem –nodes –
clcerts
```

2. Implement the following code sequence in the code of the application calling the WebService:

```
$cert = 'ścieżka_do_certyfikatu/certyfikat.pem';
$wsdl = 'https://api.santanderconsumer.pl/ProposalServiceHybrid?wsdl';
$this->Soap = new SoapClient($wsdl, array('local_cert' => $cert));
```

C#

1. Use openssl to convert the .pfx certificate into a .pem certificate. Here, too, it will be necessary to use the certificate password.

   The command that turns certyfikat.pfx into certyfikat.pem is as follows:

```
pkcs12 –in ścieżka_do_certyfikatu/certyfikat.pfx –out certyfikat.pem –nodes –clcerts
```

2. Use openssl to convert a .pem certificate into a .cer certificate.
   To change certyfikat.pem into certyfikat.cer, use the command:

```
x509 –inform PEM –in ścieżka_do_certyfikatu/certyfikat.pem –outform DER certyfikat.cer
```

3. The next step requires adding two references in the system: for the service and for the library.

4. To add a Web Reference for the service, you need to:
   - right click on the project and select "Add Service Reference";
   - in the newly opened window, select "Advanced" and then "Add Web Reference";
   - in the next window, enter the service address in the URL field, and then click "Go";
   - in the Web Reference name field, give the WebService a name, such as eRatyService, and click "Add Reference".

5. To add a Reference for the System.Security.dll library, you need to:
   - right click on the project and select "Add Reference";
   - in the newly opened window, select the ".NET" tab and search for the "System.Security" option;
   - select "System.Security" and click OK.

6. To begin with, in the code of the class that is responsible for the call, it is necessary to add a reference to the service and library.

```
using System.Security.Cryptography.X509Certificates;

using {nazwa_projektu}.eRatyService;
```

7. Use the following call code in the application itself:

```
string certPath = {ścieżka do certyfikatu/certyfikat.cer};

eRatyService.eRatyService service = new eRatyService.eRatyService();

service.ClientCertificates.Add(X509Certificate.CreateFromSignedFile(certPath));

var result =service.IsActive();
```

JAVA

In a JAVA environment, many solutions can be applied depending on the framework used. As a rule, however, there is no need to change the way WebService methods are called in the application code. Nevertheless, it is necessary to import the certificate into the so-called "TrustStore", using the JVM "keytool", and add information for the application to know where the certificate is located.

1. The .pfx format certificate sent by the Account Manager should be installed on the computer.
2. After correct installation and entering a valid password (received from the Account Manager), it will be possible to open the page with the WSDL of the service in the browser:
   https://api.santanderconsumer.pl/ProposalServiceHybrid?wsdl
3. The certificate in .pfx format should be imported into .jks (java keystore – this is the format for the JVM).
   a) Through the console, go to the directory where the JDK is installed and type:
      keytool –importkeystore –srckeystore ścieżka_do_certyfikatu/certyfikat.pfx –srcstoretype pkcs12 –destkeystore certyfikat.jks –deststoretype JKS
   b) After entering the command, you will be prompted to provide a new password twice, which must have a minimum of 5 characters and which you must enter yourself.
   c) You will then be prompted to enter the certyfikat.pfx password, and once you enter it correctly, the certificate will be added to the JVM certificate store.
4. Add the following line in the application code:
   System.setProperty("javax.net.ssl.trustStore", ścieżka_do_pliku_cacerts);.
5. If you are using a client application from which the WebService will be called, remember to add the property to the server JVM. For example, the setenv.bat entry from the Tomcat server would look like this:
   set "JAVA_OPTS=… -Djavax.net.ssl.trustStore=%CATALINA_HOME%/trustStore/certyfikat.jks"

5.2 Description of service methods

This chapter presents the specification of service methods. It should be noted that it is for illustrative purposes. The Store should implement the changes needed to integrate with the Bank's service based on the WSDL file provided.

5.2.1 IsActive

The method is used to verify that the **WebService** is working properly at a given time and is available.

Frequency of method calls

At any time in order to verify System availability.

Input

Item	Field name in WS	Section	Type	Required (YES/NO)	Description
none					

Output

Item	Field name in WS	Section	Type	Required (YES/NO)	Description
1		IsActiveResult	Complex type	YES	
2	IsCorrect	IsActiveResult	Bool	YES	Is the call valid

Item	Field name in WS	Section	Type	Required (YES/NO)	Description
3	ErrorCode	IsActiveResult	String(3)	No	Error code
4	Message	IsActiveResult	String(1024)	No	Error description
5	UserMessage	IsActiveResult	String(2048)	No	Message to be displayed to the user (does not have to involve an error)
6	LogMarker	IsActiveResult	String(32)	No	Error marker

Example of a method call

QUERY:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:san="http://santanderconsumerbank.pl/">
  <soapenv:Header/>
  <soapenv:Body>
     <san:IsActive/>
  </soapenv:Body>
</soapenv:Envelope>
```

REPLY:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
   <s:Body>
      <IsActiveResponse xmlns="http://santanderconsumerbank.pl/">
         <IsActiveResult xmlns:a="http://santanderconsumerbank.pl/objects"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
            <a:ErrorCode i:nil="true"/>
            <a:IsCorrect>true</a:IsCorrect>
            <a:LogMarker i:nil="true"/>
            <a:Message i:nil="true"/>
            <a:UserMessage i:nil="true"/>
         </IsActiveResult>
      </IsActiveResponse>
   </s:Body>
</s:Envelope>
```

5.2.2 GetApplicationState

The GetApplicationState method is used to check the status of an application.

Frequency of method calls

The method can be performed at any time to verify what the current status of a loan application is. This can be an automatic process, called once or several times a day for all active applications, or a one-time call for a specific application. With one query of this method, information on a single application or a group of applications can be obtained, while the number of applications to be checked with a single query is limited.

Input

Item	Field name in WS	Section	Type	Required (YES/NO)	Description
1		Identity	Complex type	YES	Standard object identification
2	Login	Identity	String(32)	YES	Webservice user login
3	Password	Identity	String(64)	YES	Webservice user password
4	ShopNumber	Identity	Long	YES	Point of Sale number in the Bank's system
5		ShopApplicationNumbers	Complex type	YES	List of order numbers for verification. Up to 200 order numbers are allowed in a query
6			String(64)	YES	Store order number

Output

Item	Field name in WS	Section	Type	Required (YES/NO)	Description
1		Applications	Complex type	YES	Returned list of applications
2	AgreementNumber	Applications\ApplicationData	String(12)	NO	Loan agreement number
3	ApplicationNumber	Applications\ApplicationData	String(64)	YES	Application number in the Bank's system
4	ShopApplicationNumber	Applications\ApplicationData	String(64)	YES	Store order number
5	Downpayment	Applications\ApplicationData	Double	YES	Down payment amount
6	CreditState	Applications\ApplicationData	String(64)	YES	Application status
7	TotalPrice	Applications\ApplicationData	Double	YES	Value of all goods – is used to verify whether the value of the order from the store corresponds to the value of goods in the agreement processed in the Bank
8	AgreementDate	Applications\ApplicationData	Datetime	NO	Date of loan agreement
9	ChangeDate	Applications\ApplicationData	Datetime	YES	Date of last data change on the Bank's part
10	ShopNumber	Applications\ApplicationData	Long	YES	Point of Sale number in the Bank's system
11		OperationStatus	Complex type	YES	
12	IsCorrect	OperationStatus	Bool	YES	Is the call valid
13	ErrorCode	OperationStatus	String(3)	NO	Error code
14	Message	OperationStatus	String(1024)	NO	Error description
15	UserMessage	OperationStatus	String(2048)	NO	Message to be displayed to the user (does not have to involve an error)
16	LogMarker	OperationStatus	String(32)	NO	Error marker

The list of statuses that the GetApplicationState method can return is provided in Appendix 1 to these Instructions.

Example of a method call

QUERY:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:san="http://santanderconsumerbank.pl/"
xmlns:obj="http://santanderconsumerbank.pl/objects"
xmlns:arr="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
  <soapenv:Body>
    <san:GetApplicationState>
      <san:Identity>
        <obj:Login>*******</obj:Login>
        <obj:Password>********</obj:Password>
        <obj:ShopNumber>10134</obj:ShopNumber>
      </san:Identity>
      <san:ShopApplicationNumbers>
        <arr:string>122/03/2024</arr:string>
        <arr:string>123/03/2024</arr:string>
        <arr:string>124/03/2024</arr:string>
      </san:ShopApplicationNumbers>
    </san:GetApplicationState>
  </soapenv:Body>
</soapenv:Envelope>
```

REPLY:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
   <s:Body>
      <GetApplicationStateResponse xmlns="http://santanderconsumerbank.pl/">
         <GetApplicationStateResult xmlns:a="http://santanderconsumerbank.pl/objects"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
            <a:OperationStatus>
               <a:ErrorCode i:nil="true"/>
               <a:IsCorrect>true</a:IsCorrect>
               <a:LogMarker i:nil="true"/>
               <a:Message i:nil="true"/>
               <a:UserMessage i:nil="true"/>
            </a:OperationStatus>
            <a:Applications>
               <a:ApplicationData>
                  <a:AgreementDate i:nil="true"/>
                  <a:AgreementNumber i:nil="true"/>
                  <a:ApplicationNumber>603450</a:ApplicationNumber>
                  <a:ChangeDate>2024-03-04T09:50:42.313738+01:00</a:ChangeDate>
                  <a:CreditState>BANK</a:CreditState>
                  <a:Downpayment>0</a:Downpayment>
                  <a:ShopApplicationNumber>123/03/2024</a:ShopApplicationNumber>
                  <a:ShopNumber>10134</a:ShopNumber>
                  <a:TotalPrice>3000</a:TotalPrice>
               </a:ApplicationData>
               <a:ApplicationData>
                  <a:AgreementDate i:nil="true"/>
                  <a:AgreementNumber i:nil="true"/>
                  <a:ApplicationNumber>603451</a:ApplicationNumber>
                  <a:ChangeDate>2024-03-04T09:50:42.313738+01:00</a:ChangeDate>
                  <a:CreditState>KLIENT</a:CreditState>
                  <a:Downpayment>0</a:Downpayment>
                  <a:ShopApplicationNumber>124/03/2024</a:ShopApplicationNumber>
                  <a:ShopNumber>10134</a:ShopNumber>
                  <a:TotalPrice>2500</a:TotalPrice>
               </a:ApplicationData>
               <a:ApplicationData>
                  <a:AgreementDate>2024-03-04T07:50:42.313738+01:00</a:AgreementDate>
                  <a:AgreementNumber>649112230871</a:AgreementNumber>
                  <a:ApplicationNumber>603452</a:ApplicationNumber>
                  <a:ChangeDate>2024-03-04T09:50:42.313738+01:00</a:ChangeDate>
                  <a:CreditState>WYDAJ_TOWAR</a:CreditState>
                  <a:Downpayment>0</a:Downpayment>
                  <a:ShopApplicationNumber>122/03/2024/a:ShopApplicationNumber>
                  <a:ShopNumber>10134</a:ShopNumber>
                  <a:TotalPrice>1500</a:TotalPrice>
               </a:ApplicationData>
            </a:Applications>
         </GetApplicationStateResult>
      </GetApplicationStateResponse>
   </s:Body>
</s:Envelope>
```

5.2.3 GetFinancialData

The GetFinancialData method is used to simulate a loan, that is, to calculate its financial parameters. The simulation result obtained is the basis for preparing information for the Customer about the cost of the loan.

Frequency of method calls

The method should be called every time you open the simulator page and change the parameters available there.

Input

Item	Field name in WS	Section	Type	Required (YES/NO)	Description
1		Identity	Complex type	YES	Standard object identification
2	Login	Identity	String(32)	YES	Webservice user login
3	Password	Identity	String(64)	YES	Webservice user password
4	ShopNumber	Identity	Long	YES	Number of the Point of Sale for which the simulation is called
5		Security	Complex type	YES	
6	IPNumber	Security	String(39)	YES	IP number of the computer on which the simulation was performed
7	PortNumber	Security	Int	YES	Client port information
8	BrowserID	Security	String(256)	YES	Information on the browser in which the simulation was performed
9	OperationDate	Security	Datetime	YES	Time from the customer’s browser
10	BrowserFingerprintHash	Security	String(64)	YES	Code calculated according to the algorithm described at amiunique.org
11		Simulation	Complex type	YES	Input data for calculating loan parameters
12	ProductNumber	Simulation	String(14)	YES	Identifier of the installment product for which the simulator query is created – received from the Account Manager
13	InstalmentsNumber	Simulation	Int	YES	Number of installments
14	Downpayment	Simulation	Double(10,2)	YES	Down payment
15	TotalPrice	Simulation	Double(11,2)	YES	The sum of prices of the goods, together with the cost of shipping, which can also be covered by the loan.

Output

Item	Field name in WS	Section	Type	Required (YES/NO)	Description
1		SimulationResult	Complex type	YES	Results of the simulation performed
2	CreditAmount	SimulationResult	Double	YES	Total amount of the loan
3	InterestRate	SimulationResult	Double	YES	Interest rate (% per annum)
4	APR	SimulationResult	Double	YES	APR (Annual Percentage Rate)
5	TotalCreditCost	SimulationResult	Double	YES	Total cost of the loan
6	InstalmentAmount	SimulationResult	Double	YES	Monthly installment
7	InstalmentsNumber	SimulationResult	Int	YES	Number of installments
8	AdjustmentInstalmentAmount	SimulationResult	Double	YES	Last adjustment installment (as part of the application process – value ignored for the purpose of performing simulations)
9	TotalAmountToPay	SimulationResult	Double	YES	Total amount to pay
10	CommissionAmount	SimulationResult	Double	YES	Commission
11	InterestAmount	SimulationResult	Double	YES	Interest (amount)
12	TotalAdditionalFee	SimulationResult	Double	YES	Total amount of additional fee in the agreement (as part of the application process – value ignored for the purpose of performing simulations)
13	Downpayment	SimulationResult	Double	YES	Down payment (amount)
14	Clause	SimulationResult	String	YES	Content of the representative example
15		OperationStatus	Complex type	YES	
16	IsCorrect	OperationStatus	Bool	YES	
17	ErrorCode	OperationStatus	String(3)	NO	Error code
18	Message	OperationStatus	String(1024)	NO	Error description
19	UserMessage	OperationStatus	String(2048)	NO	Message to be displayed to the user (does not have to involve an error)
20	LogMarker	OperationStatus	String(32)	NO	Error marker

Example of a method call

QUERY:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:san="http://santanderconsumerbank.pl/"
xmlns:obj="http://santanderconsumerbank.pl/objects">
   <soapenv:Header/>
   <soapenv:Body>
      <san:GetFinancialData>
         <san:Identity>
            <obj:Login>************</obj:Login>
            <obj:Password>************</obj:Password>
            <obj:ShopNumber>99995</obj:ShopNumber>
         </san:Identity>
         <san:Security>
            <obj:BrowserFingerprintHash>29ff458b17b93ab32ef8eb105b6e770c</obj:BrowserFingerprintHash>
            <obj:BrowserID>Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/72.0.3626.119 Safari/537.36</obj:BrowserID>
            <obj:IPNumber>10.66.40.165</obj:IPNumber>
            <obj:OperationDate>2024-03-01T14:26:27</obj:OperationDate>
            <obj:PortNumber>8181</obj:PortNumber>
         </san:Security>
         <san:Simulation>
            <obj:Downpayment>0</obj:Downpayment>
            <obj:InstalmentsNumber>24</obj:InstalmentsNumber>
            <obj:ProductNumber>107</obj:ProductNumber>
            <obj:TotalPrice>3874.4</obj:TotalPrice>
         </san:Simulation>
      </san:GetFinancialData>
   </soapenv:Body>
</soapenv:Envelope>
```

REPLY:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
   <s:Body>
      <GetFinancialDataResponse xmlns="http://santanderconsumerbank.pl/">
         <GetFinancialDataResult xmlns:a="http://santanderconsumerbank.pl/objects" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
            <a:OperationStatus>
               <a:ErrorCode i:nil="true"/>
               <a:IsCorrect>true</a:IsCorrect>
               <a:LogMarker i:nil="true"/>
               <a:Message i:nil="true"/>
               <a:UserMessage i:nil="true"/>
            </a:OperationStatus>
            <a:SimulationResult>
               <a:APR>11.12</a:APR>
               <a:AdjustmentInstalmentAmount>0</a:AdjustmentInstalmentAmount>
               <a:Clause>Dla kredytu na zakup towarów i usług, oferowanego przez Santander Consumer Bank S.A. z siedzibą we Wrocławiu, zawartego na 10 miesięcy, wyliczenia dla przykładu reprezentatywnego na dzień 23.07.2019 r. są następujące… Decyzja o warunkach i przyznaniu kredytu z uwzględnieniem oceny aktualnej sytuacji Klienta, zostanie podjęta przez Santander Consumer Bank S.A.</a:Clause>
               <a:CommissionAmount>38.74</a:CommissionAmount>
               <a:CreditAmount>3874.4</a:CreditAmount>
               <a:Downpayment>0</a:Downpayment>
               <a:InstalmentAmount>138.37</a:InstalmentAmount>
               <a:InstalmentsNumber>24</a:InstalmentsNumber>
               <a:InterestAmount>653.12</a:InterestAmount>
               <a:InterestRate>10</a:InterestRate>
               <a:TotalAdditionalFee>0</a:TotalAdditionalFee>
               <a:TotalAmountToPay>4566.26</a:TotalAmountToPay>
               <a:TotalCreditCost>691.86</a:TotalCreditCost>
            </a:SimulationResult>
         </GetFinancialDataResult>
      </GetFinancialDataResponse>
   </s:Body>
</s:Envelope>
```

5.6 Operation errors

If the operation call fails, an error message (SOAP exception) will be returned. It contains additional error details, in the form of a simple XML document, whose faultString node contains two subnodes:
CODE – containing the error code,
DESCRIPTION – containing a textual description of the error.

Error example:

```
<soap:Body>
<soap:Fault>
<faultCode>
      soap:Server
</faultCode>
<faultString>
      &lt;KOD&gt;13&lt;/KOD&gt;&lt;OPIS&gt;&lt;/OPIS&gt;
</faultString>
<detail/>
</soap:Fault>
</soap:Body>
```

General messages that can be returned by service methods

Code	Name	GetFinancialData	GetApplicationState
1	Invalid parameter <parameter name>	+	+
2	Login error	+	+
3	No store privileges	+	+
4	Internal error in the eRaty system	+	+
5	Invalid product type	+	+
6	Sales channel not available in the store	+	+
8	No store in the system	+	+
9	No <method name> method privileges	+	+
21	No active installment products	+	
22	Invalid installment product identifiers	+	
23	Validity of the loan product expired	+	
35	Simulation error, loan limit exceeded	+	
36	Simulation error, too low amount of loan	+	
52	Invalid number of installments for the product	+	
54	Down payment too low	+	
101	Invalid application number		+
102	No application privileges		+
199	General validation error <ewentualny_opis>	+	+
301	Application parameters inconsistent with link parameters	+	

603	Maintenance break	+	+
605	Link not found	+	
701	No application identifier		+
702	No application at SCB		+
703	SCB application number inconsistent with epoint		+
705	Invalid product parameters	+	
712	The application has already been authorized	+	
722	No customer computer data available	+	
740	Value of goods inconsistent with the cost of shipping	+	

6. Additional options

6.1 Information on the Store's website

Banners

Any Store offering the Bank's installment payment option should provide its Customers with information on the indicated payment method, along with a description of the process of purchasing installment goods/services and information on credit brokerage. This information can be described, for example, in a dedicated section on the Store's website or in the form of a banner redirecting the Customer to the Bank's website.
Ready-to-use information/advertising banners are available at https://www.eraty.pl/do-pobrania/banery-ogolne/. The banner posted on the Store's website should be linked to the Bank's website address: https://www.santanderconsumer.pl/raty-jak-kupic, where it is explained in a few simple steps how to use the Bank's installment payment option when shopping in the online store.

To adjust the descriptions to the appearance (structure) of the Store's website, you can use the functionality described below.

Example for PHP language

Site placement: https://www.santanderconsumer.pl/raty-jak-kupic
using the PHP **file_get_contents();** function

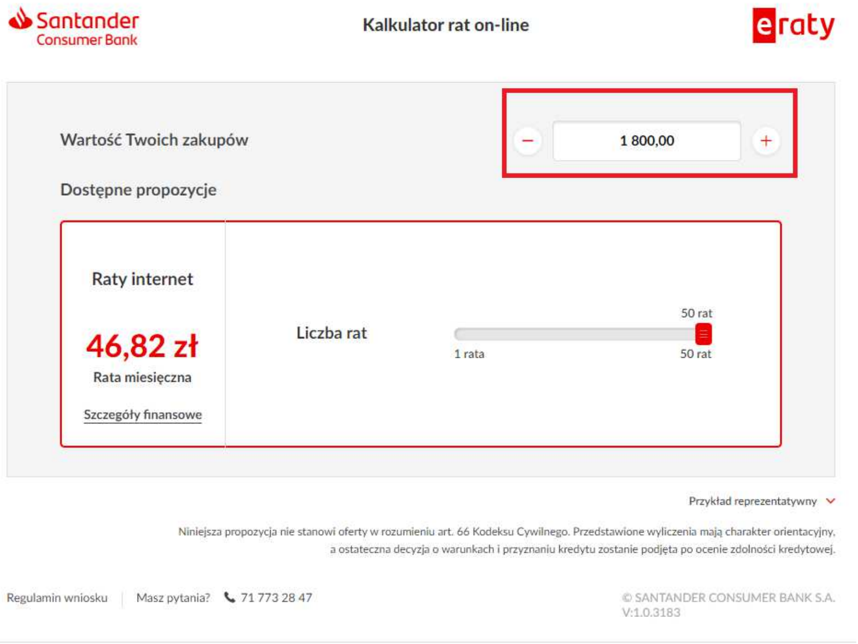
```
<?PHP
echo file_get_contents('https://www.santanderconsumer.pl/raty-jak-kupic');
?>
```

In addition at https://www.eraty.pl/do-pobrania/landing-page/ there is a downloadable and ready-to-use description of the process of applying for a loan for the purchase of goods and services in the Bank, as well as a description of the benefits of using such a form of payment.

6.2 Installment simulator

The installment simulator is a tool provided to online stores to allow their Customers to simulate the calculation of the amount and cost of loans for various parameters, i.e. the number of installments or the price of goods, even before placing an order.

The simulator is not an integral part of the application, so the Customer's navigation to the simulator is not the beginning of the application process for a loan for the purchase of goods and services.



Santander Consumer Bank

Kalkulator rat on-line

eraty

Wartość Twoich zakupów

− 1 800,00 +

Dostępne propozycje

Raty internet

46,82 zł

Rata miesięczna

Szczegóły finansowe

Liczba rat

Przykład reprezentatywny

Niniejsza propozycja nie stanowi oferty w rozumieniu art. 66 Kodeksu Cywilnego. Przedstawione wyliczenia mają charakter orientacyjny, a ostateczna decyzja o warunkach i przyznaniu kredytu zostanie podjęta po ocenie zdolności kredytowej.

Regulamin wniosku | Masz pytania? ☎ 71 773 28 47

© SANTANDER CONSUMER BANK S.A.
V:1.0.3183

HOW TO CALL THE SIMULATOR FROM THE STORE LEVEL

The simulator is called via a GET request directly from the online store.

It should open in a new browser window and calculate the loan installment amount for the given input parameters. A good solution for the installment simulator window is to open it with JavaScript. This will allow you to determine the size of the newly opened window and its other properties. With this solution, the installment simulation will not cover up the entire store window, but only a small part of it, making it easier for the Customer to navigate.

The structure of the query and the URL calling the simulator is as follows:

https://wniosek.eraty.pl/symulator/oblicz/PARAMETR1/WARTOSC1/PARAMETR2/WARTOSC2

A typical call has the following form (all mandatory parameters are listed):

https://wniosek.eraty.pl/symulator/oblicz/numerSklepu/*/typProduktu/*/wartoscTowarow/*

where parameters mean:

– numerSklepu – Store number in the Bank's system,
– typProduktu – always takes the value "0" (zero),
– wartoscTowarow – value of the goods for which the simulation is to be performed.

In the simulator call, you can also indicate an optional parameter with a default number of installments, as in the following example calling the execution of a simulation on the test Store 99995 for goods with a value of PLN 1000.99 and 12 installments.

https://wniosek.eraty.pl/symulator/oblicz/numerSklepu/99995/typProduktu/0/wartoscTowarow/1000.99/liczbaRat/12

WHERE TO PLACE THE LINK TO THE SIMULATOR ON THE STORE'S WEBSITE

On the Store's website, the link to the installment simulator is usually available under the "Calculate installment" button. It can be placed both on the product description pages, as well as in the shopping cart preview or on the order confirmation and payment method selection page.
Logos and buttons are available for download at https://www.eraty.pl/do-pobrania/kredyt-na-zakup-towarow-i-uslug/

In the <head>...</head> section of the website, you can put the following code:

```
<script
type="text/javascript">
function PoliczRate(koszyk) {

      window.open('https://wniosek.eraty.pl/symulator/oblicz/numerSklepu/
      */typProduktu/0/wartoscTowarow/'+koszyk, 'Policz_rate',
     'width=630,height=500,directories=no,location=no,menubar=no,resizable=yes,scrollba
     rs=yes,status=no,toolbar=no');
}
</script>
```

Installment purchase is possible when the price of the goods is higher than PLN 100. Below is an example (in PHP) of controlling the simulator's availability depending on the price of the goods:

```
<?PHP
if ($cena_produktu >= 100) {
      echo 'TEN PRODUKT KUPISZ U NAS NA R A T Y<br />
      <a OnClick="PoliczRate(' . $cena_produktu . ');" style="cursor:pointer;">
      <img src="https://www.eraty.pl/images/bannery/oblicz_white.png" title="Oblicz
      ratę" alt="Oblicz ratę" border="0" />
      </a
>'; }
else {
      echo 'Zakup na raty dostępny jest <b>od 100 zł</b> wartości produktu lub
      całego zamówienia.';
}
```

```
?>
```

6.3 Link generator

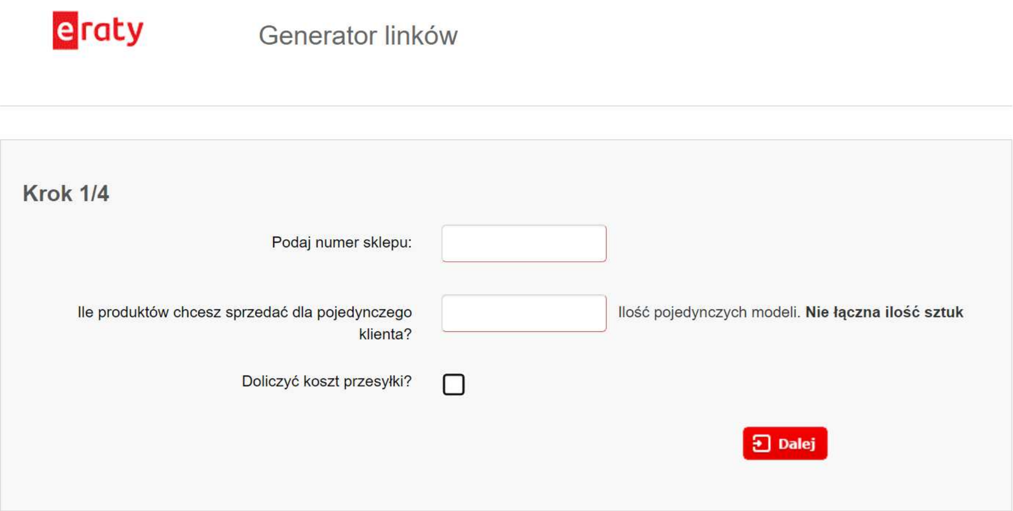
The link generator is a tool used by an online store to complete individual purchase transactions. It allows you to prepare for the Customer a dedicated link to the application for the purchase of selected goods.

The link generator is used in cases where:

- full integration of the online store with the eRaty system is not possible or not required; or
- the Store wants to occasionally provide the installment payment option; or
- the price of the goods is determined with the Customer individually, for example, in a telephone conversation or online auction; or
- it is necessary for the Customer to re-submit the order.

ABOUT THE PROCESS USING THE LINK GENERATOR

The link generator is available to the merchant in Panel Sklep or at https://generator.eraty.pl.



Generator linków

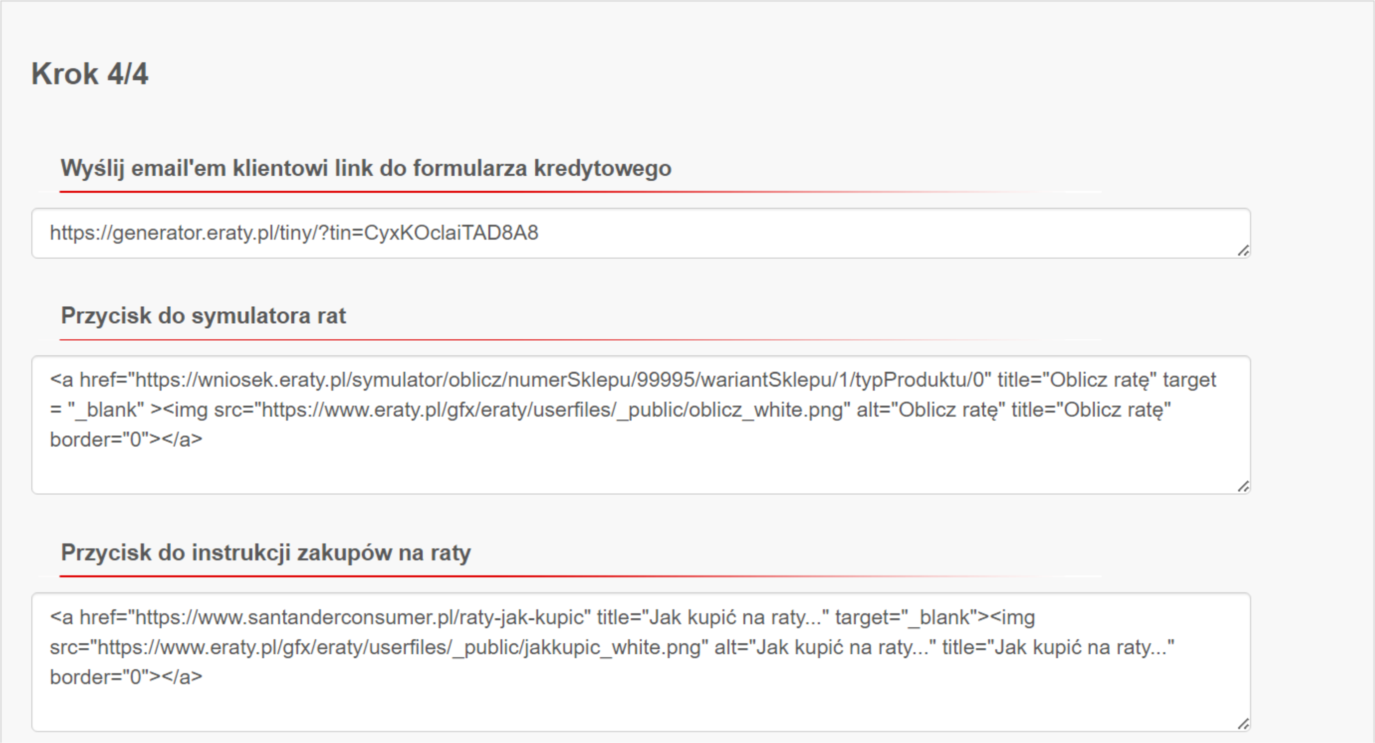
Krok 1/4

Podaj numer sklepu:

Ile produktów chcesz sprzedać dla pojedynczego klienta? Ilość pojedynczych modeli. **Nie łączna ilość sztuk**

Doliczyć koszt przesyłki? □

In order to generate a link, it is necessary to fill in all the required data for each step in the form. In the last step, the generator will return the required links.



Krok 4/4

Wyślij email'em klientowi link do formularza kredytowego

```
https://generator.eraty.pl/tiny/?tin=CyxKOclaiTAD8A8
```

Przycisk do symulatora rat

```
<a href="https://wniosek.eraty.pl/symulator/oblicz/numerSklepu/99995/wariantSklepu/1/typProduktu/0" title="Oblicz ratę" target
= "_blank" ><img src="https://www.eraty.pl/gfx/eraty/userfiles/_public/oblicz_white.png" alt="Oblicz ratę" title="Oblicz ratę"
border="0"></a>
```

Przycisk do instrukcji zakupów na raty

```
<a href="https://www.santanderconsumer.pl/raty-jak-kupic" title="Jak kupić na raty..." target="_blank"><img
src="https://www.eraty.pl/gfx/eraty/userfiles/_public/jakkupic_white.png" alt="Jak kupić na raty..." title="Jak kupić na raty..."
border="0"></a>
```

When using the link generator for an online auction, the generated HTML codes for the installment purchase instructions button and for the installment simulator button should be copied and pasted into the HTML code of the auction. After the auction is completed, send the Customer an e-mail with the generated link to the application – a link from the field: “Send the customer an e-mail with the link to the loan application form”.
If the link generator is used for standard purchases, then after completing the forms, all you need to do is send the Customer an e-mail with the generated link.
The generated link will redirect the Customer to the loan application of the eRaty system, where they will be able to complete the process of financing their purchase.

7. How to carry out the process of integration with the Bank

The integration process is complex and requires several steps, both on the Store’s and the Bank’s part.

What a Partner should do in order to integrate their Store with the Bank

Before starting the integration process, the Partner should:

- sign a cooperation agreement with the Bank,
- notify their Account Manager of the desire to integrate and the planned method of integration.

What the Account Manager should do after conclusion of the cooperation agreement

After conclusion of the cooperation agreement, the Account Manager requests:

- parameterization in the Bank’s Point of Sale application;
- granting personal privileges to Panel Sklep to Store employees;

- granting authorization for WebService communication with the Bank and forwarding the required certificate to the Store (optional, depending on the planned method of integration).

What data and information does the Account Manager provide to the Store

To begin integration works, the Account Manager submits the following data to the Store:

- Point of Sale number at the Bank;
- numbers and names of loan products and the financial terms assigned to them;
- Panel Sklep login credentials for Store employees;
- a unique login and password for the WebService (optional, depending on the planned method of integration);
- data necessary for installing the required certificate.

Is it possible to conduct integration testing

Once all the necessary integration works are completed, testing can be conducted. Testing takes place on both the Bank's part and the integrated Store's part.

Once the technical integration is completed on the Store's part, its authorization by the Bank is required. To do so, send an e-mail to: wsparcie@santanderconsumer.pl, with carbon copy to the Account Manager.

The e-mail should contain:
– in the subject line: "Test request"
– in the content: the URL of the integrated store, company name, phone number and Point of Sale number.

About testing on the Bank's part

Testing on the Bank's part includes technology infrastructure:

- visibility and correctness of the link to the installment simulator for goods with a price higher than PLN 100,
- visibility and correctness of the link to the installment simulator in the order summary,
- ability to choose payment by the Bank's installment loan from the list of available payment methods in the Store,
- proper redirection to the loan application form after selecting installment payment.

In addition, at this stage of the integration process, the Account Manager is required to verify:

- correctness of the marketing content regarding the Bank's offerings posted on the Store's website, and
- Store's fulfillment of its information obligation towards the Customer, including posting of appropriate representative examples or information on credit brokerage.

Testing on the Bank's part does not include:

- the entire loan application process, but only the part up to the point of checking consents and statements, and
- graphic presentation of the Store's offering and banners with information about the Bank's installment payment option.

About testing on the Store’s part

Testing on the Store’s part should include the same steps as on the Bank’s part, and in addition:

- ability to log into Panel Sklep and ability to view submitted loan applications,
- verification of correctness and completeness of e-mail correspondence from the Bank informing about the status of loan applications,
- verification of correctness of WebService communication, including correctness of the API connection, the feedback messages received from the Bank and their correct interpretation (in particular, this applies to messages related to the release of goods or their withholding) – optional test, depending on the selected method of integration.

Integration verification takes place in the Bank’s production environment. Initial tests can be performed on Point of Sale No. 99995, which is in test mode in the Bank’s system.

Proper tests are conducted on the Point of Sale number indicated in the appendix to the cooperation agreement with the Bank. Before proceeding to testing:
– The Account Manager requests the activation of the test mode for a given Point of Sale,
– The Store changes the Point of Sale number in the integration settings.

What if there are questions or problems at the integration testing stage

If you have any questions or problems that arise during the integration testing phase, please contact the Bank via e-mail.
The e-mail sent to wsparcie@santanderconsumer.pl with a carbon copy to the Account Manager should include:
– in the subject line: “eHP integration tests –*[Store URL*]”
– in the content: a detailed description of the problem/concern, including screenshots of error messages, code snippets and the URL of the integrated Store, Partner’s name, phone number and Point of Sale number.

Additionally, for WebService integration:
– the full URL to which the request is directed,
– information on whether an access certificate has been installed,
– full content of the request sent to the WebService (without the password visible),
– WebService’s full response.

When is it possible to start the process in production

Production launch of the process and provision of loan for the purchase of goods and services as a form of financing the purchase in the Store is possible after successful completion of integration tests on the Bank’s and the Store’s part. After verification of all formal issues, the Account Manager requests the change of parameterization of the Point of Sale to production mode and informs the Store about the date of sales launch.

8. Appendices

Appendix 1 – List of Application Statuses

ID statusu	Status of the application as seen in the Panel Sklep application	Application status returned by the GetApplicationState method
-127	Rejestracja nr wniosku (status niewidoczny w Panel Sklep)	Klient(-127)
-105	Klient zamówił kontakt z Contact Center (status niewidoczny w Panel Sklep)	Bank (-105)
-104	Klient uzupełnił dane kontaktowe na wniosku (status niewidoczny w Panel Sklep)	Klient (-104)
-101	Klient porzucił wniosek. Bank podejmie próbę kontaktu z klientem. (status niewidoczny w Panel Sklep)	Bank (-101)
-100	Klient uzupełnił dane kontaktowe na wniosku (status niewidoczny w Panel Sklep)	Klient (-100)
1	Klient złożył wniosek	Bank
2	Wniosek przekazany do kontaktu z klientem	Bank
3	Wniosek przyjęty do weryfikacji	Bank
4	Weryfikacja warunków kredytu - kontakt z klientem	Bank
6	Wniosek w trakcie dodatkowej weryfikacji	Bank
7	Klient otrzymał zgodę na zawarcie umowy	Bank
8	Brak zgody na zawarcie umowy	Odmowa
9	Wniosek anulowany	Odmowa
10	Klient otrzymał dokumentację kredytową	Klient
11	Dokumenty klienta w takcie w weryfikacji	Bank
12	Umowa zawarta i skierowana do uruchomienia	Wydaj_towar
14	Umowa uruchomiona – środki z kredytu przelane do kontrahenta	Wydaj_towar
17	Rezygnacja klienta	Odmowa
19	Umowa zarchiwizowana	Wydaj_towar
21	Potwierdzenie dostępności towarów	Sklep
22	Klient otrzymał indywidualną zgodę na zawarcie umowy	Bank
24	Wniosek anulowany z powodu braku towaru	Odmowa
29	Klient uzupełnia brakujące dokumenty	Klient
32	Wniosek oczekuje na weryfikację	Bank
38	Brak zgody na zawarcie umowy po ponownej weryfikacji	Odmowa
42	Błąd techniczny na wniosku	Bank
45	Wniosek w trakcie potwierdzania tożsamości klienta i sprawdzania dokumentów	Bank
46	Klient otrzymał zgodę na zawarcie umowy - wymagane zaświadczenie	Bank
49	Oczekiwanie na dane z przelewu klienta	Klient
50	Wymagany ponowny przelew	Klient
51	Dane z przelewu w trakcie weryfikacji	Bank
52	Wniosek anulowany na rzecz nowego - wymagana ponowna autoryzacja	Bank
54	Dane z przelewu zweryfikowane negatywnie	Bank
55	Wniosek skierowany do dodatkowej weryfikacji	Bank
67	Umowa zamknięta z powodu odstąpienia	Zakonczona
69	Klient zaakceptował zmienione warunki oferty	Bank
70	Klient otrzymał zgodę na zawarcie umowy - zmienione warunki	Bank
71	Klient otrzymał zgodę na zawarcie umowy - zmienione warunki i wymagane zaświadczenie	Bank
74	Potwierdzenie tożsamości przez przelew	Klient
77	Dane z przelewu zweryfikowane pozytywnie	Bank
81	Brak zgody na zawarcie umowy - kontakt z klientem	Bank

82	Wymagane załączenie skanu dokumentu tożsamości	Klient
106	Zamówione towary częściowo niedostępne - kontakt z klientem	Bank
115	Podpisanie umowy kodem SMS nie powiodło się	Odmowa
117	Wymagane załączenie skanu dokumentów	Klient
118	Skany dokumentów załączone	Bank
119	Klient otrzymał kod SMS do podpisania umowy	Klient
120	Umowa podpisana kodem SMS	Bank